

Εργαστήριο 3 - Άσκηση - Ανάλυση

Εκφώνηση: Δημιουργείτε την εφαρμογή «Αγορά Σπιτιού» σε Java Swing με χρήση NetBeans ακολουθώντας τις παρακάτω οδηγίες.

1. Η εφαρμογή θα σχεδιασθεί σε ένα εξωτερικό υποδοχέα JFrame, ο οποίος θα έχει τίτλο «Αγορά Σπιτιού» και θα οριστεί με την έξοδο του να κλείνει και η εφαρμογή.
 - ❖ Στο JFrame θα τοποθετήσετε ένα JPanel με ιδιότητες:
 - Background: επιλογής σας – προσέξτε την αντίθεση με το background του JFrame
 - Border: TitledBorder με τίτλο «Στοιχεία Δανείου» με font επιλογής σας.
 - Επίσης, στο JPanel θα τοποθετήσετε τρία JLabel και τρία JTextField όπως φαίνονται στην εικόνα.
 - ❖ Στο JFrame θα τοποθετήσετε ένα δεύτερο JPanel με ιδιότητες:
 - Background: επιλογής σας – προσέξτε την αντίθεση με το background του JFrame
 - Border: TitledBorder με τίτλο «Διαθέσιμα Σπίτια» με font επιλογής σας.
 - Επίσης, στο JPanel θα τοποθετήσετε τρία radiobuttons (μην ξεχάσετε να ορίσετε το ButtonGroup τους) και δύο Labels:
 - Το πρώτο RadioButton θα έχει τις παρακάτω ιδιότητες:
 - Θα περιέχει μόνο εικόνα χωρίς κείμενο. Στην ιδιότητα icon θα ορίσουμε το αρχείο εικόνας ChatsworthHouse.jpg .
 - Όταν πατιέται (pressedIcon) ή επιλέγεται (selectedIcon) ή φωτίζεται προσωρινά (rolloverIcon) η εικόνα θα αλλάζει σε (ChatsworthHousePressed.jpg).
 - Border EtchedBorder με πλαίσιο (BorderPainted)
 - Να είναι αρχικά επιλεγμένη (ιδιότητα selected).
 - Το δεύτερο RadioButton θα έχει τις παρακάτω ιδιότητες:
 - Θα περιέχει μόνο εικόνα (queen_house.jpg)
 - Όταν πατιέται ή επιλέγεται ή φωτίζεται προσωρινά η εικόνα θα αλλάζει σε (queen_housePressed.jpg)
 - Border EtchedBorder με πλαίσιο (BorderPainted)
 - Το τρίτο RadioButton θα έχει τις παρακάτω ιδιότητες:
 - Θα περιέχει μόνο εικόνα (white_house.jpg)
 - Όταν πατιέται ή επιλέγεται ή φωτίζεται προσωρινά η εικόνα θα αλλάζει σε (white_housePressed.jpg)
 - Border EtchedBorder με πλαίσιο (BorderPainted)
 - Δύο Labels όπως φαίνεται στην εικόνα.
 - ❖ Επίσης, στο JFrame θα τοποθετήσετε ένα button και δύο Label όπως φαίνεται στην εικόνα.

Στη συνέχεια θα προγραμματίσετε τη συμπεριφορά των συστατικών της διεπιφάνειας σας. Αναλυτικά:

- Ορίστε στα radiobutton το γεγονός **ItemStateChanged** ώστε όταν επιλέγεται το κουμπί να αλλάζει η εικόνα του κουμπιού στη μορφή "...pressed" και να ορίζεται τιμή του σπιτιού σε
 - 3.000.000 για το 1^ο σπίτι,
 - 5.000.000 για το 2^ο σπίτι και
 - 4.000.000 για το 3^ο σπίτι.
- Ορίστε στα radiobutton τα γεγονότα **MouseEntered** και **MouseExited** ώστε όταν μπαίνει ο δείκτης του ποντικιού να αλλάζει προσωρινά η αντίστοιχη τιμή στην ετικέτα, ενώ όταν βγαίνει να επιστρέφει στην προηγούμενη κατάσταση.
- Ορίστε στο κουμπί «Υπολογισμός Δόσης» το γεγονός **ActionPerformed** ώστε να υπολογίζει την κατάλληλη δόση ανάλογα με τις τιμές που έχουν τα συστατικά των δύο πάνελ. Το αποτέλεσμα να το εμφανίζει στην κατάλληλη ετικέτα.
- Ορίστε συντομεύσεις για τα παρακάτω συστατικά:
 - Κεφάλαιο – το αγγλικό κουμπί 'Κ',
 - Επιτόκιο – το αγγλικό κουμπί 'Ε',
 - Διάρκεια – το αγγλικό κουμπί 'Δ',
 - Υπολογισμός Δόσης – το αγγλικό κουμπί 'Υ',
 - 1ο radiobutton – τον αριθμό 1,
 - 2ο radiobutton – τον αριθμό 2,
 - 3ο radiobutton – τον αριθμό 3.

Μη ξεχάσετε να κάνετε τις απαραίτητες ενέργειες για να φαίνεται η κατάλληλη υπογράμμιση στα 4 πρώτα συστατικά που θα οριστεί συντόμευση ώστε να ενημερώνεται ο χρήστης.

Σημείωση: Μέθοδοι που πιθανώς θα σας φανούν χρήσιμες:

- `TextField.setText(string);`
- `TextField.getText();`
- `String.valueOf(double);`
- `Double.valueOf(string);`
- `Integer.valueOf(string);`
- `Πεδίο_String.contentEquals(string)`

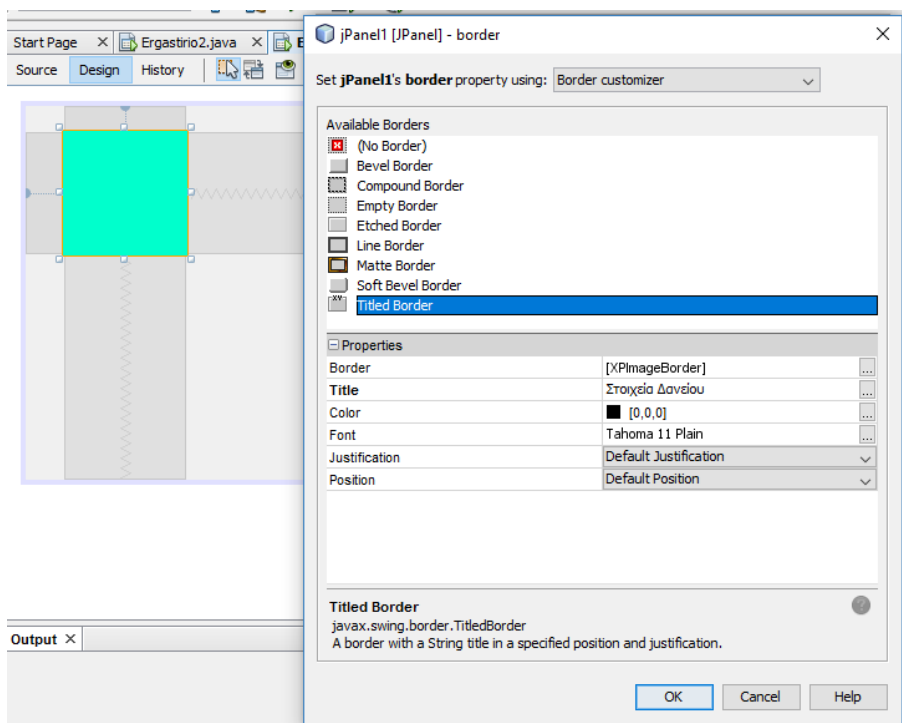
Ανάλυση και εξήγηση λύσης

Βήμα 1: Δημιουργούμε ένα νέο project στο NetBeans με όνομα Askisi5.

Βήμα 2: Δημιουργούμε ένα νέο Java αρχείο τύπου JFrame Form με όνομα Ergastirio3.

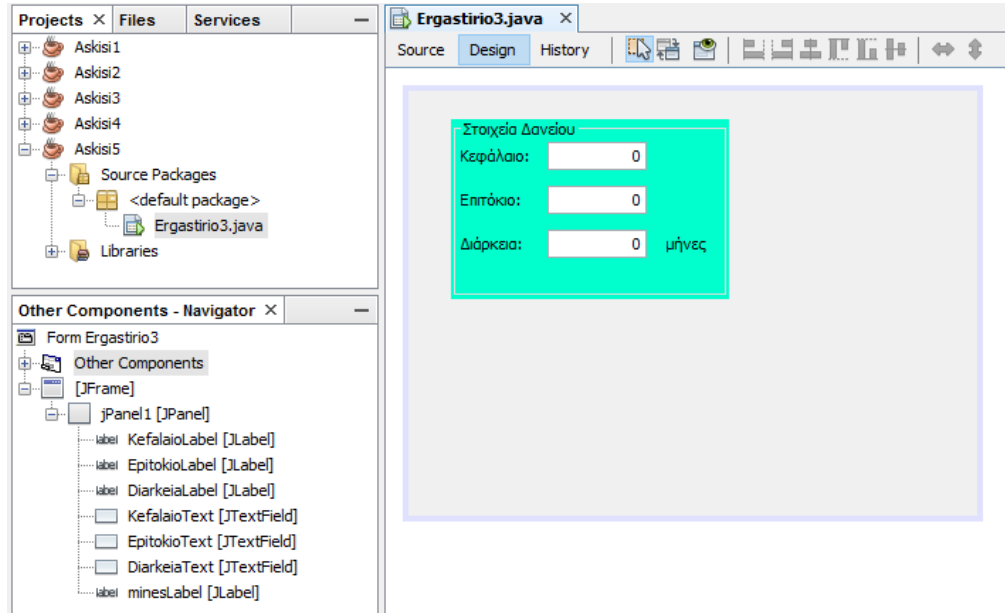
Βήμα 3: Ορισμός Ιδιοτήτων στο JFrame. Ορίζουμε ως τίτλο του JFrame το «Αγορά Σπιτιού».

Βήμα 4: Τοποθετούμε στον καμβά ένα JPanel. Το JPanel είναι ένα Swing Container, και το βρίσκουμε στην εργαλειοθήκη της Java Swing στα Swing Container. Από τις ιδιότητες του JPanel αλλάζουμε το χρώμα του background – επιλέξτε ένα απαλό χρώμα που όμως να είναι διακριτό από το γκρι χρώμα του JFrame. Στη συνέχεια αλλάζουμε το περίγραμμα (border) του JPanel σε titled border. Από τις ιδιότητες του JPanel επιλέγουμε τις «...» δεξιά στην ιδιότητα *border*. Από το παράθυρο που θα ανοίξει από τα «Available Borders» επιλέγουμε το «Titled Border». Στις ιδιότητες (Properties) του «Titled Border» (στο κάτω μέρος του παραθύρου) ορίζουμε τις προτιμήσεις μας. Έτσι, στο «Title» ορίζουμε ως τίτλο του JPanel «Στοιχεία Δανείου».



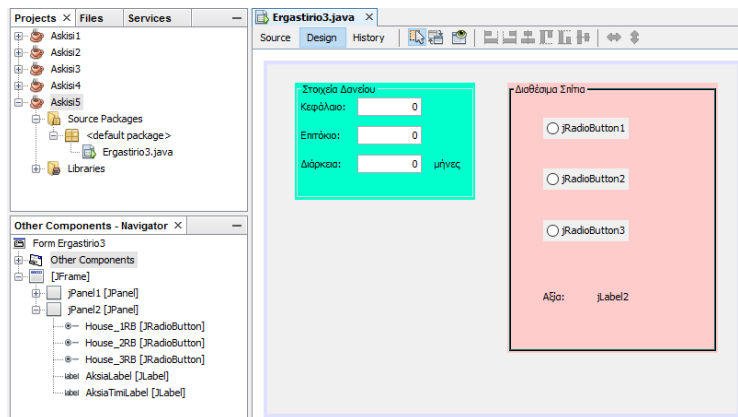
Εικόνα 1: Ορισμός Titled Border

Βήμα 5: Τοποθετούμε τέσσερα JLabel και τρία TextField στο JPanel. Ορίζουμε τα ονόματα μεταβλητών και το κείμενο τους όπως φαίνεται στην Εικόνα 2. Επίσης, στα τρία JTextField ορίζουμε στην ιδιότητα *HorizontalAlignment* την τιμή *RIGHT* για δεξιά στοίχιση του περιεχομένου JTextField.



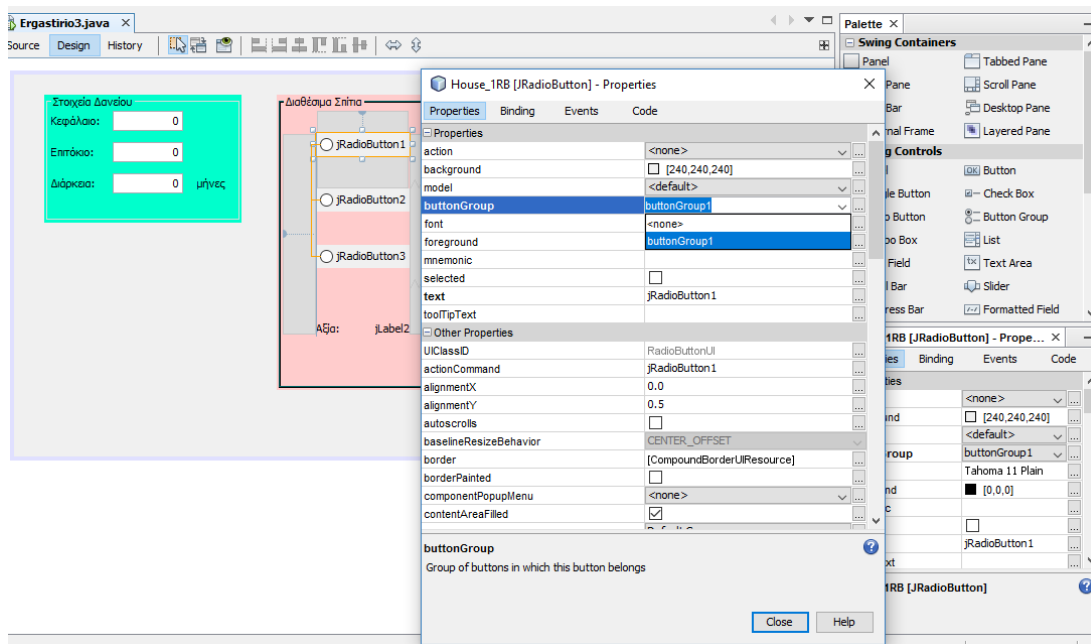
Εικόνα 2: Το πρώτο JPanel

Βήμα 6: Τοποθετούμε στον καμβά ένα δεύτερο JPanel. Από τις ιδιότητες του JPanel αλλάζουμε το χρώμα του background και το περίγραμμα (border) του JPanel σε titled border. Στο «Title» ορίζουμε ως τίτλο του JPanel «Διαθέσιμα Σπίτια». Επίσης, προσθέτουμε στο JPanel τρία JRadioButtons, δύο JLabels και ένα ButtonGroup. Και τα τρία συστατικά τα βρίσκουμε στα Swing Controls. Το buttonGroup είναι αόρατο στον καμβά. Στο navigator τοποθετείται στα «other components», εκεί όπου τοποθετούνται όλα τα αόρατα συστατικά. Το buttongroup θα το χρησιμοποιήσουμε για να ομαδοποιήσουμε σε μία ομάδα τα τρία JRadioButtons – θα το δούμε στο επόμενο βήμα.



Εικόνα 3: Η αρχική εικόνα του δεύτερου JPanel

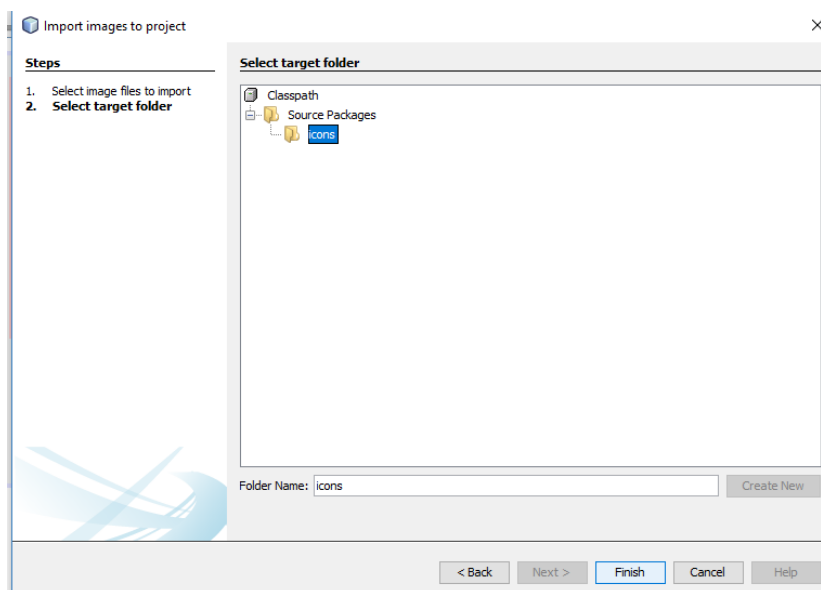
Βήμα 7: Τα RadioButtons έχουν την ιδιαιτερότητα, όταν οριστούν να ανήκουν στην ίδια ομάδα, να είναι επιλεγμένο μόνο ένα RadioButton από όλη την ομάδα. Με το ButtonGroup δηλώνουμε ομάδες RadioButton. Στο προηγούμενο βήμα προσθέτοντας το ButtonGroup δημιουργήθηκε η ομάδα με όνομα buttonGroup1. Για να ορίσουμε τα τρία RadioButtons να είναι στην ίδια ομάδα, ορίζουμε να έχουν και τα τρία για την ιδιότητα buttonGroup την τιμή buttonGroup1, όπως φαίνεται στην Εικόνα 4.



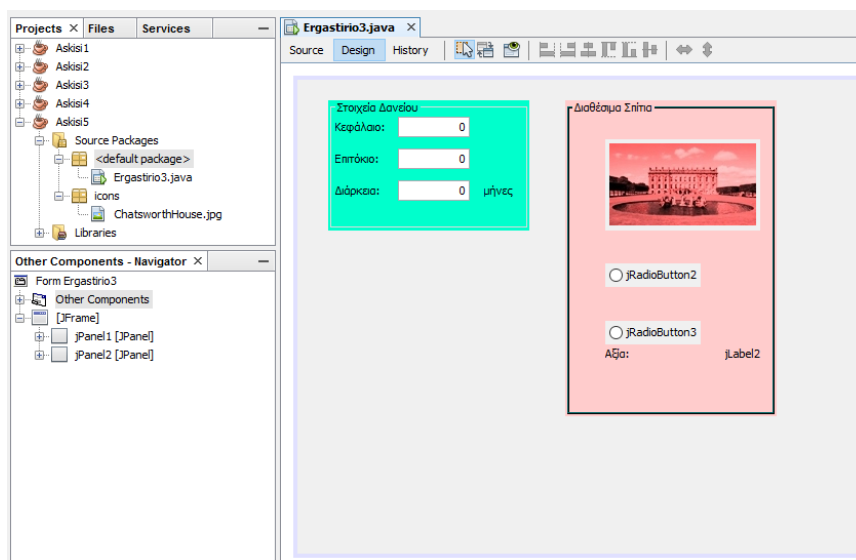
Εικόνα 4: Ομαδοποίηση των τριών jRadioButton

Βήμα 8: Τα radioButton δεν θα έχουν περιεχόμενο κείμενο αλλά εικόνα. Έτσι, θα σβήσουμε το κείμενο από τα radioButton, δηλαδή από την ιδιότητα text θα σβήσουμε το jRadioButton1 για το πρώτο κουμπί. Στη συνέχεια θα ορίσουμε στην ιδιότητα icon την εικόνα του αρχείου «ChatsworthHouse.jpg».

Τα έξι αρχεία εικόνων μπορείτε να τα αποθηκεύσετε σε ένα προσωρινό κατάλογο. Τα αρχεία εικόνων θα τα εισάγουμε στο project καθώς θα ορίζουμε τις ιδιότητες. Έτσι, για να ορίσουμε στην ιδιότητα icon το κατάλληλο αρχείο εικόνας αλλά και να εισάγουμε την εικόνα στο project, θα επιλέξουμε από τις ιδιότητες του πρώτου radioButton τις «...» δεξιά της ιδιότητας icon. Στο παράθυρο που θα ανοίξει θα επιλέξουμε το κουμπί «Import To Project». Στη συνέχεια θα βρούμε το αρχείο εικόνας «ChatsworthHouse.jpg», θα το επιλέξουμε και θα επιλέξουμε το κουμπί «Next» για να πάμε στο επόμενο βήμα. Μπορούμε την εικόνα να την εισάγουμε στο src κατάλογο του project ή να δημιουργήσουμε ένα νέο κατάλογο, όπως φαίνεται στην Εικόνα 5, και επιλέγουμε «Finish». Κλείνουμε το παράθυρο και παρατηρούμε (Εικόνα 6) ότι το radioButton περιέχει την εικόνα αλλά και την εμφάνιση του καταλόγου icons ως package στο project.

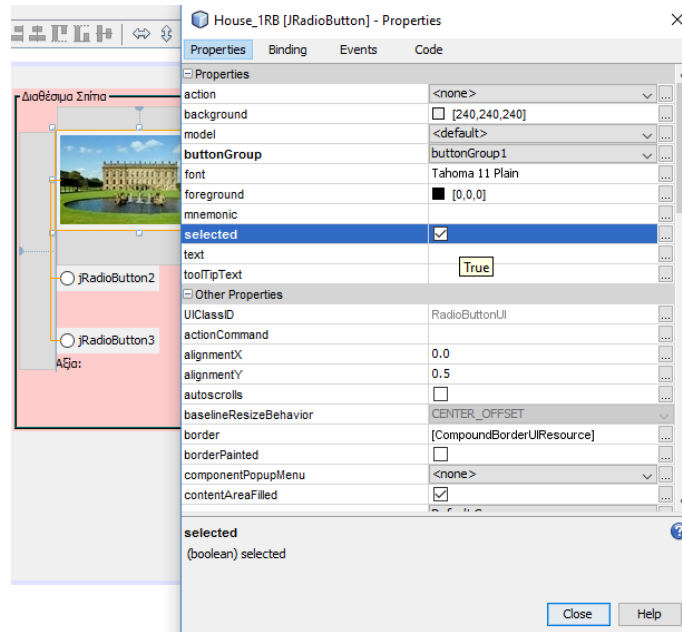


Εικόνα 5: Δημιουργία του καταλόγου icons



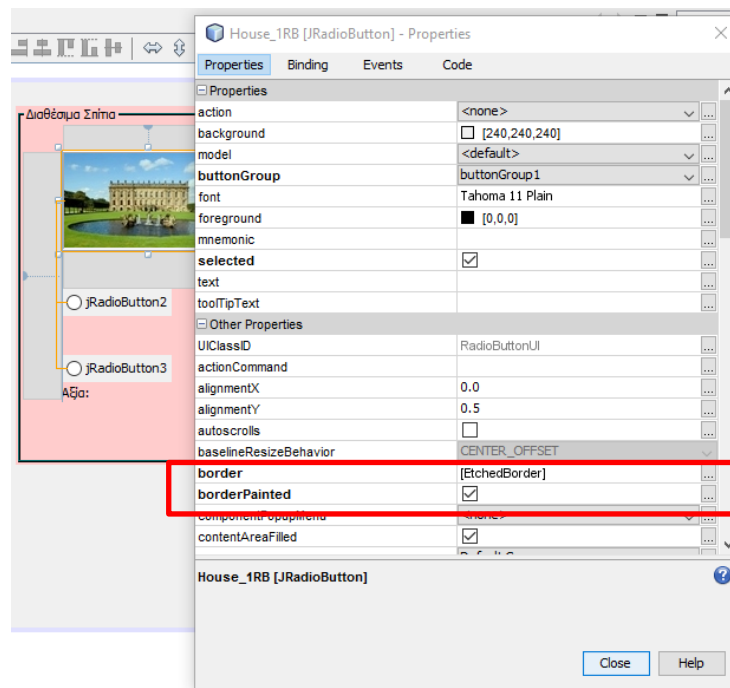
Εικόνα 6: Το JRadioButton περιέχει εικόνα

Στη συνέχεια ορίζουμε στην ιδιότητα ***pressedIcon*** την εικόνα του αρχείου εικόνας «ChatsworthHousePressed.jpg». Η συγκεκριμένη ιδιότητα ορίζει, ότι όταν θα επιλέγει ο χρήστης το radiobutton με το αριστερό πλήκτρο του ποντικιού θα αλλάζει η εικόνα του κουμπιού σε αυτή που περιέχει το αρχείο εικόνας. Την ίδια εικόνα ορίζουμε και στις ιδιότητες ***rolloverIcon***, η εικόνα του κουμπιού θα αλλάζει στην εικόνα του αρχείου εικόνας που ορίσαμε στην ιδιότητα όποτε περνάμε το ποντίκι μέσα από το κουμπί χωρίς να πατήσουμε κάποιο πλήκτρο του ποντικιού, και ***selectedIcon***, η οποία ορίζει ποια εικόνα θα εμφανίζεται στο κουμπί όταν αυτό θα είναι το επιλεγμένο της ομάδας – ως επιλεγμένο ορίζεται αυτό που έχει δηλωθεί αρχικά να έχει την ιδιότητα ***selected*** (Εικόνα 7) ή το τελευταίο που θα επιλέξουμε με το αριστερό πλήκτρο του ποντικιού.



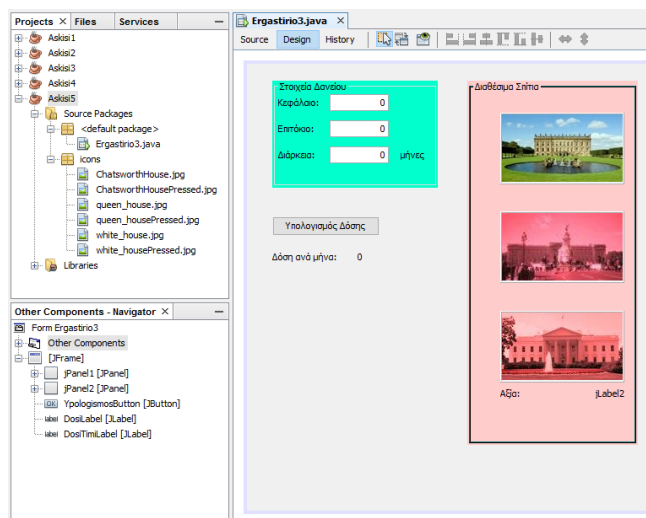
Εικόνα 7: Ιδιότητα selected

Τέλος, ορίζουμε το radiobutton να έχει την κατάλληλη τιμή στην ιδιότητα border και στην ιδιότητα borderPainted, όπως φαίνεται στην Εικόνα 8.



Εικόνα 8: Ιδιότητες border και borderPainted

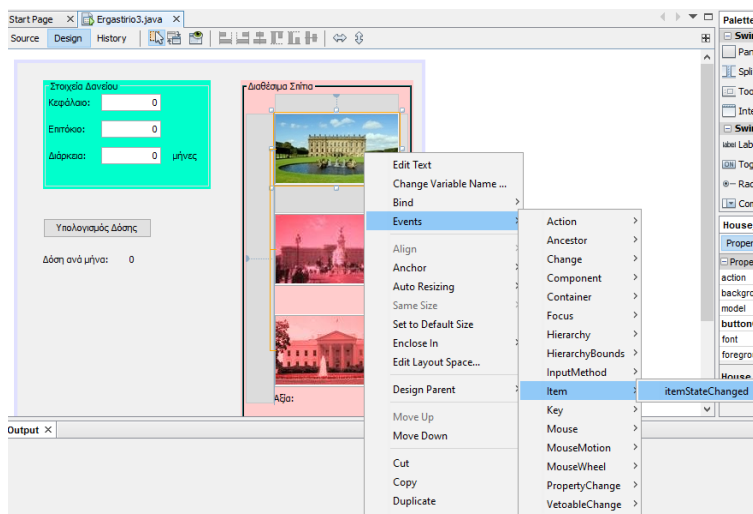
Επαναλαμβάνουμε το βήμα 8 και για τα άλλα δύο radioButton σύμφωνα με την εκφώνηση. Ολοκληρώνουμε τη σχεδίαση της διεπαφής προσθέτοντας ένα jButton και άλλα δύο jLabel στον καμβά κάτω από το πρώτο jPanel όπως φαίνεται στην Εικόνα 9.



Εικόνα 9: Ολοκλήρωση της σχεδίασης της διεπαφής

Βήμα 9: Σε αυτό το βήμα θα προγραμματίσουμε τι θα συμβαίνει όταν ο χρήστης της εφαρμογής επιλέγει ένα από τα τρία σπίτια (radioButtons) με το αριστερό πλήκτρο του ποντικιού. Πολλά event μπορούν να προγραμματιστούν για το συγκεκριμένο γεγονός. Το πιο κατάλληλο για να διαχειριστούμε radiobuttons είναι το *item->itemStateChanged*.

Έτσι, επιλέγουμε να προγραμματίσουμε για το πρώτο radiobutton το συγκεκριμένο event (Εικόνα 10). Θα πρέπει να γίνουν δύο ενέργειες όταν εκτελείται το συγκεκριμένο γεγονός. Πρώτον να εμφανίζεται στην ετικέτα το ποσό του σπιτιού και δεύτερον σε μία global μεταβλητή price να αποθηκεύεται η τιμή του σπιτιού. Αυτή τη global μεταβλητή θα τη χρησιμοποιήσουμε αργότερα. Στην Εικόνα 11, φαίνεται ο κώδικας των τριών radiobutton για το συγκεκριμένο γεγονός ενώ στη γραμμή 281 δηλώνεται η global μεταβλητή. Τέλος, αλλάζουμε το text της ετικέτας AksiaTimiLabel σε 3.000.000 για να αντιστοιχεί στην τιμή του πρώτου σπιτιού (Εικόνα 12).



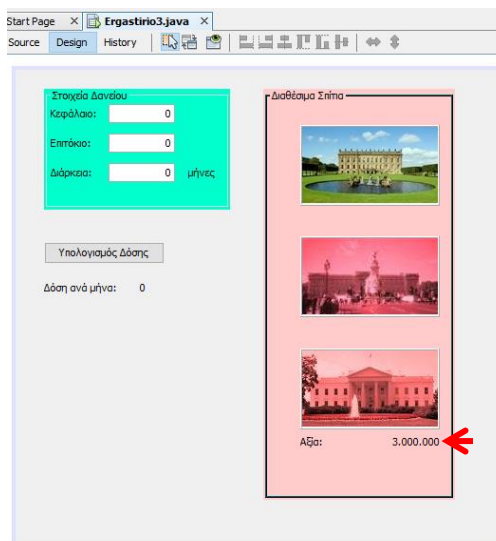
Εικόνα 10: Item – itemStateChanged γεγονός


```

Start Page x Ergastirio3.java x
Source Design History
231 private void House_1RBItemStateChanged(java.awt.event.ItemEvent evt) {
232     AksiaTimiLabel.setText("3.000.000");
233     price = "3.000.000";
234 }
235
236 private void House_2RBItemStateChanged(java.awt.event.ItemEvent evt) {
237     AksiaTimiLabel.setText("5.000.000");
238     price = "5.000.000";
239 }
240
241 private void House_3RBItemStateChanged(java.awt.event.ItemEvent evt) {
242     AksiaTimiLabel.setText("4.000.000");
243     price = "4.000.000";
244 }
245
246 /**
247  * @param args the command line arguments
248  */
249 public static void main(String args[]) { ...31 lines }
280
281 String price = "3.000.000";
282 // Variables declaration - do not modify

```

Εικόνα 11: Ο κώδικας των τριών radiobutton για το γεγονός itemStateChanged



Εικόνα 12: Το text του JLabel AksiaTimiLabel παίρνει την τιμή του πρώτου radiobutton

Βήμα 10: Όταν ο χρήστης θα μεταφέρει το δείκτη του ποντικιού μέσα σε ένα radiobutton χωρίς να πατήσει κάποιο πλήκτρο θα πρέπει να αλλάζει το text της ετικέτας AksiaTimiLabel στην τιμή του συγκεκριμένου σπιτιού που απεικονίζεται στο radiobutton. Αυτό γίνεται προγραμματίζοντας το event MouseEntered σε κάθε ένα από τα τρία radiobutton, όπως φαίνεται στην Εικόνα 13. Ενώ όταν θα μεταφέρει το δείκτη του ποντικιού έξω από το radiobutton θα πρέπει να αλλάζει το text της ετικέτας AksiaTimiLabel στην τιμή του επιλεγμένου (selected) σπιτιού της ομάδας. Αυτό γίνεται προγραμματίζοντας το event MouseExited σε κάθε ένα από τα τρία radiobutton, όπως φαίνεται στην Εικόνα 14. Για να

υλοποιήσουμε αυτό το γεγονός χρησιμοποιήσουμε τη μεταβλητή *price*, στην οποία είναι αποθηκευμένη η τιμή του επιλεγμένου σπιτιού.

```

261 private void House_1RButtonEntered(java.awt.event.MouseEvent evt) {
262     AksiaTimiLabel.setText("3.000.000");
263 }
264
265 private void House_2RButtonEntered(java.awt.event.MouseEvent evt) {
266     AksiaTimiLabel.setText("5.000.000");
267 }
268
269 private void House_3RButtonEntered(java.awt.event.MouseEvent evt) {
270     AksiaTimiLabel.setText("4.000.000");
271 }
272

```

Εικόνα 13: Ο κώδικας των τριών radiobutton για το γεγονός MouseEntered

```

282 private void House_1RButtonExited(java.awt.event.MouseEvent evt) {
283     AksiaTimiLabel.setText(price);
284 }
285
286 private void House_2RButtonExited(java.awt.event.MouseEvent evt) {
287     AksiaTimiLabel.setText(price);
288 }
289
290 private void House_3RButtonExited(java.awt.event.MouseEvent evt) {
291     AksiaTimiLabel.setText(price);
292 }
293
294 /**
295  * @param args the command line arguments

```

Εικόνα 14: Ο κώδικας των τριών radiobutton για το γεγονός MouseExited

Βήμα 11: Τώρα θα προγραμματίσουμε το γεγονός ActionPerformed του κουμπιού “Υπολογισμός Δόσης” για να βρούμε τη δόση δανείου που θα πληρώσει ο πελάτης σύμφωνα με τις τέσσερις παραμέτρους, δηλαδή την αξία σπιτιού, το κεφάλαιο που έχει ο πελάτης (τα χρήματα που έχουν), το επιτόκιο δανεισμού και τον αριθμό των δόσεων που επιθυμεί ο πελάτης. Στην Εικόνα 15, παρουσιάζεται ο κώδικας υπολογισμού της δόσης.

Στη γραμμή 303, η μεταβλητή *kefalaioD* χρησιμοποιείται για να «κρατάει» το κεφάλαιο που έχει δηλώσει ο πελάτης ότι έχει στο TextField *KefalaioText*. Στο TextField έχει εισαχθεί η τιμή ως κείμενο. Θεωρούμε ότι το *KefalaioText* δεν είναι κενό και περιέχει αριθμό. Σε περίπτωση που ο αριθμός έχει γραφτεί σε μορφή «N.NNN.NNN», δηλαδή με χρήση της τελείας για διαχωρισμό των ψηφίων τότε με την μέθοδο *replace* αφαιρούμε την τελεία. Τέλος πριν αποθηκευτεί το ποσό του κεφαλαίου που δηλώθηκε στο *KefalaioText* στη μεταβλητή *kefalaioD* μετατρέπεται σε *double*.

Οι επόμενες τρεις γραμμές του κώδικα είναι παρόμοιες για το επιτόκιο του δανείου, τη διάρκεια της δόσης σε μήνες και την αξία του σπιτιού.

Στη γραμμή 308 βρίσκουμε το δάνειο που χρειάζεται ο πελάτης. Αν το δάνειο είναι αρνητικό σημαίνει ότι ο πελάτης έχει τα λεφτά για να αγοράσει το σπίτι και έτσι εμφανίζουμε κατάλληλο μήνυμα (γραμμή 310) στην ετικέτα *EtiketaEnimerosis* που προσθέτουμε στο JFrame σε κατάλληλο σημείο (Εικόνα 16). Αλλιώς, υπολογίζουμε τον τόκο (γραμμή 313), βρίσκουμε το συνολικό ποσό (δάνειο + τόκος) (γραμμή 314), και αν η διάρκεια έχει οριστεί μεγαλύτερη από το μηδέν, βρίσκουμε (γραμμή 318) και εμφανίζουμε τη δόση (γραμμή 319). Αν η διάρκεια δεν έχει οριστεί μεγαλύτερη του μηδενός εμφανίζουμε κατάλληλο μήνυμα (γραμμή 316).

```

300
301
302 private void YpologismosButtonActionPerformed(java.awt.event.ActionEvent evt) {
303     double kefalaioD = Double.valueOf(KefalaioText.getText().replace(".", ""));
304     double epitokioD = Double.valueOf(EpitokioText.getText().replace(".", ""));
305     double diarkeiaD = Double.valueOf(DiarkeiaText.getText().replace(".", ""));
306     double axiaD = Double.valueOf(price.replace(".", ""));
307
308     double daneio = axiaD - kefalaioD;
309     if (daneio <= 0)
310         EtiketaEnimerosis.setText("Δεν χρειάζεσαι δάνειο!");
311     else {
312         EtiketaEnimerosis.setText("");
313         double tokos = (daneio * epitokioD)/100;
314         double daneiometoko = daneio + tokos;
315         if (diarkeiaD <= 0)
316             DosiTimiLabel.setText("Λάθος διάρκεια");
317         else {
318             double dosi = daneiometoko/diarkeiaD;
319             DosiTimiLabel.setText(String.valueOf(dosi));
320         }
321     }
322 }
323

```

Εικόνα 15: Ο κώδικας υπολογισμού της δόσης



Εικόνα 16: Εμφάνιση κατάλληλου μηνύματος αν ο πελάτης δεν χρειάζεται δάνειο.

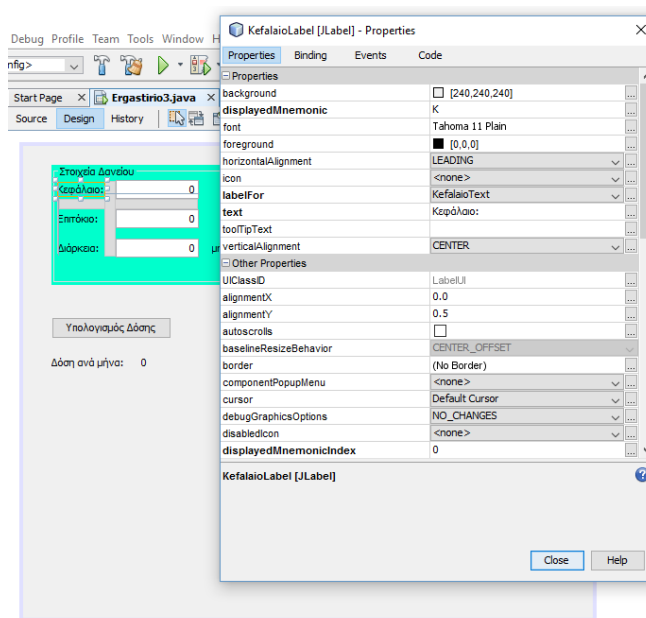
Βήμα 12: Στο τελευταίο βήμα της άσκησης θα ορίσουμε τις συντομεύσεις. Με αυτές θα μπορεί ο χρήστης να εκτελέσει κάποιες ενέργειες με συνδυασμό πλήκτρων από το πληκτρολόγιο.

Έτσι, για να ορίσουμε συντόμευση σε ένα TextField θα πρέπει να οριστεί η συντόμευση σε μία ετικέτα (JLabel). Όπως φαίνεται στην Εικόνα 17, για να ορίσουμε τη συντόμευση ALT+K για το

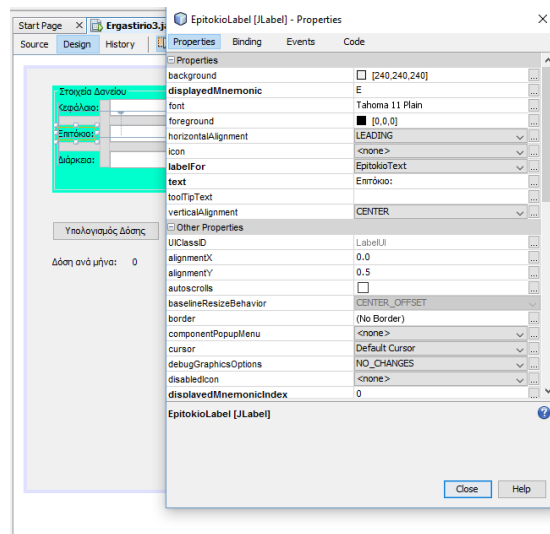
KefalaioText θα πρέπει να ορίσουμε τρεις ιδιότητες της ετικέτας *KefalaioLabel* που προσδιορίζει το *KefalaioText*. Την ιδιότητα *displayedMnemonic*, όπου ορίζουμε το γράμμα (Κ) που θα συνδυάσουμε με το πλήκτρο ALT (default) – προσοχή αν ορίσουμε ελληνικό γράμμα θα πρέπει το πληκτρολόγιο να είναι ελληνικό όταν θα χρησιμοποιούμε την συντόμευση. Την ιδιότητα *labelFor*, όπου ορίζουμε τη μεταβλητή του TextField που θέλουμε να μεταφερθεί ο cursor με το συνδυασμό των πλήκτρων ALT+K. Τέλος, την ιδιότητα *displayedMnemonicIndex*, όπου ορίζουμε ποιο γράμμα θα υπογραμμιστεί στο JLabel για να δείξουμε στο χρήστη πως θα χρησιμοποιήσει τη συντόμευση – ορίζουμε έναν αριθμό, ξεκινώντας από το μηδέν προσδιορίζοντας το γράμμα του κειμένου της ετικέτας που θέλουμε να υπογραμμιστεί. Δηλαδή, για να υπογραμμιστεί το πρώτο γράμμα χρησιμοποιούμε το μηδέν, για το δεύτερο το ένα κτλ. Με παρόμοιο τρόπο ορίζουμε τις συντομεύσεις και στα άλλα δύο TextFields (Εικόνα 18 και Εικόνα 19).

Για το JButton *ΥπολογισμοςButton* αρκεί να ορίσουμε τις ιδιότητες *mnemonic* και *displayedMnemonicIndex* όπως φαίνεται στην Εικόνα 20.

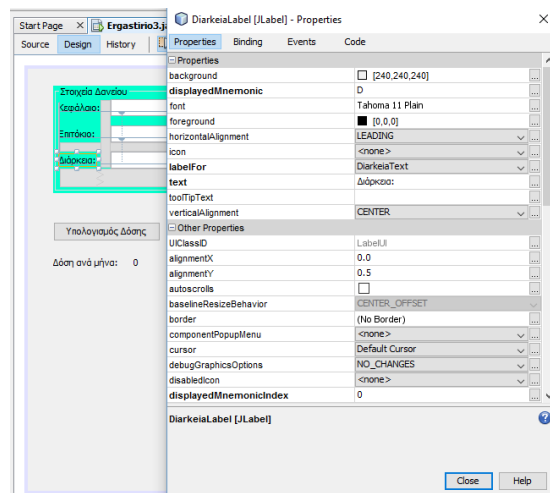
Τέλος, για τα radioButton που περιέχουν μόνο εικόνα δεν χρειάζεται να οριστεί η ιδιότητα *displayedMnemonicIndex*. Αρκεί να οριστεί κατάλληλα η ιδιότητα *mnemonic* όπως φαίνεται στην Εικόνα 21, για το πρώτο radioButton.



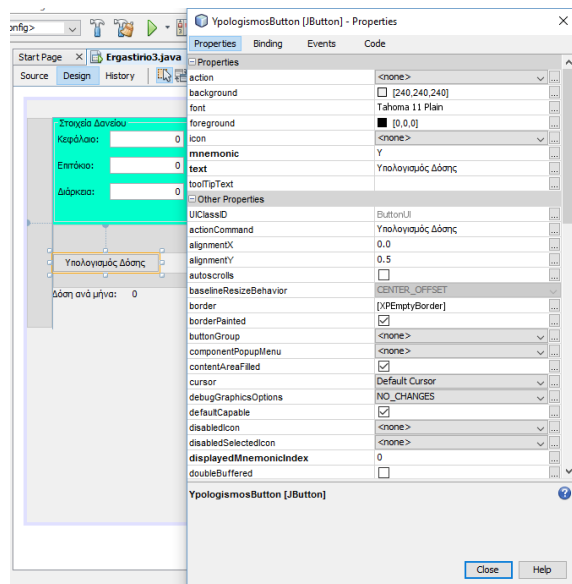
Εικόνα 17: Ορισμός συντομεύσεων για το KefalaioText



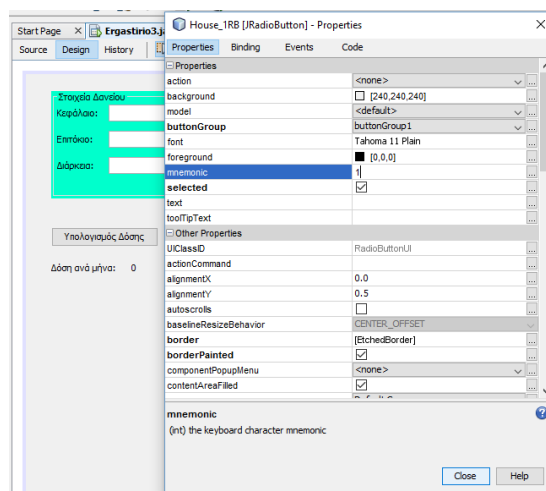
Εικόνα 18: Ορισμός συντομεύσεων για το EritokioText



Εικόνα 19: Ορισμός συντομεύσεων για το DiarkeiaText



Εικόνα 20: Ορισμός συντομεύσεων για το YpologismosButton



Εικόνα 21: Ορισμός συντομεύσεων για το House_1RB